САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Математико-механический факультет

Кафедра информатики

АБДУРАХМАНОВА Кямиля Фуадовна

СИСТЕМА ОЦЕНКИ ПОРАЖАЕМОСТИ КРИТИЧНЫХ ДОКУМЕНТОВ НА ОСНОВЕ ВЕРОЯТНОСТНО – РЕЛЯЦИОННОЙ МОДЕЛИ: СОЦИОИНЖЕНЕРНЫЕ АТАКИ

Дипломная работа

по специальности 010300 “Фундаментальные информатика и информационные технологии”

Научный руководитель – доктор физико-математических наук,
профессор А. Л. Тулупьев

Студент: [подпись]

Научный руководитель: [подпись]

Работа представлена на кафедру

“___” _____ 2015/6 г.

Заведующий кафедрой: [подпись]

Санкт-Петербург

2016

SAINT-PETERSBURG STATE UNIVERSITY

Mathematics and Mechanics Faculty

the Department of Computer Science

ABDURAKHMANOVA Kiamilia

CRITICAL ASSESSMENT SYSTEM affection documents based on probabilistic relational model : socio-engineering attacks

Bachelor's Thesis

the specialty 010300 "Basic Computer Science and Information Technology"

Scientific supervisor – Doctor of Physical and Mathematical Sciences ,

Professor A. L. Tulupyev

Student: [подпись]

Scientific supervisor: [подпись]

Работа представлена на кафедру

"___" _____ 2015/6 г.

Head of Department: [подпись]

Saint - Petersburg

2016

Оглавление

ВВЕДЕНИЕ

В настоящее время информационные ресурсы используются повсеместно, кроме того, вместе экспоненциальным увеличением объема информации, растет число атак на систему.

В 2014 году компания Arbor Networks провела ряд исследований на увеличение количества программно-технических атак в сети Интернет, результатом которого было увеличение числа вдвое за год [1]. Вследствие чего, актуальна проблема защиты систем, а также целостности документов компании, хранящихся в ней.

Однако начали набирать популярность атаки, направленные не на саму систему, а на ее пользователей, так называемые социоинженерные атаки. В данном случае, злоумышленники хотят получить доступ к данным, не пытаясь взломать систему, а воздействуя на пользователей – к примеру, учитывая их психологические особенности и используя слабости. Исследователи склоняются к тому, технической безопасности уделяется внимание и принимаются меры по её обеспечению, и наиболее уязвимым элементом информационных систем остается человек [2].

Следует учитывать, что при социоинженерных атаках обнаружение утечки данных для компании происходит после нанесения урона. Поэтому остро стоит проблема анализа поражаемости документов в системе, а также ее пользователей, на предмет социоинженерной атаки. Данная работа посвящена оценке поражаемости данных посредством таких атак.

Целью данной работы было создание системы оценки поражаемости критичных документов, имитирующей атаку на систему и отображающей исходы атаки и «слабые места» в системе.

В данной работе были поставлены следующие задачи:

- Используя метод Н. В. Хованова [3] определения общей модели измерения ценности, расширить модели, входящие в последовательность «информационная система – персонал – критичные документы»
- Разработать алгоритм имитации социоинженерной атаки злоумышленника на пользователей системы
- Разработать алгоритм вывода мер предосторожности (защиты): разработать алгоритм поиска самых уязвимых пользователей, хостов и документов
- Реализовать вышеуказанные алгоритмы в программных модулях системы анализа защищенности пользователей от социоинженерных атак

I. I ГЛАВА

I.1 Введение

Данная глава посвящена рассмотрению проблемы защищенности пользователей информационных систем. Представлены различные подходы к анализу данной проблемы. Отмечены преимущества и недостатки каждого из подходов, обоснована значимость проблемы. Кроме того, в данной главе дано краткое описание подходов к автоматизированному анализу программно-технической защищенности информационных систем.

I.2 Информационная безопасность: о методах атак

Существует различные подходы к классификации к классификации угроз информационной безопасности. В [4] описывается подробная классификация угроз, согласно источнику угрозы, его положению, привлекательности атаки, степени возможного повреждения, вероятности успеха, характеру атаки, и другим параметрам. В [5] приведена классификация источников угроз. Обобщая эти классификации, мы сформировать классификацию атак по источнику угроз:

1. Стихийные бедствия
2. Обусловленные техническими средствами
    a. Внутренние
    b. Внешние
3. Обусловленные действиями субъектов
    a. Внутренние
    b. Внешние

Согласно [4], внутренние источники атак наиболее опасны, так как пользователи имеют доступ к техническим средствам и конфиденциальной информации, и их действия могут принести значительный урон организации. В [2] сообщается, что статьях говорится о том, что высокую ответственность

за безопасность и конфиденциальность данных в ИС несет сам пользователь. Однако, авторы этих статей не выделяют отдельную категорию угроз, когда третья лица воздействуют на пользователей ИС с целью получить доступ к конфиденциальным данным, извлечь выгоду, нарушить работу предприятия или нарушить нормальную работу ИС.

Такое воздействие называется «социоинженерными атаками». В стандартах от Microsoft [6] сообщается, что в таких атаках злоумышленник использует приемы социальной инженерии с целью получения доступа к закрытой информации и ресурсам, и приводится классификация таких атак:

- сетевые атаки,
- телефонные атаки,
- поиск информации в мусоре, офисных отходах,
- персональные подходы,
- обратная социотехника.

Злоумышленники, использующие методы социальной инженерии, обычно преследуют типичные цели: деньги, доступ к ресурсам и информации компании, порча имущества, и для достижения своих целей используют лень, доверие и энтузиазм сотрудников. Злоумышленники пытаются убедить сотрудников предоставить им ресурсы, или же совершить необходимые им действия руками сотрудников. Иногда атакуемые предприятия используются как плацдарм для последующих атак. Нынешний рост киберпреступности, и в частности – социоинженерных атак на малые и средние предприятия, свидетельствуют о том, что социоинженерные становятся всё более серьезной и актуальной проблемой. [6]

Социальная инженерия является мощным средством, которое может нанести колоссальный ущерб компании. Оценить суммарный урон, сделанный

социоинженерными атаками, тяжело, так как пользователи часто не сообщают о том, что были атакованы. Это происходит из-за того, что пользователь, совершая действия, навязанные злоумышленниками, руководится противозаконными мотивами, или действует ради наживы. Однако, крупнейшие атаки почтовых червей использовали приемы социальной инженерии, чтобы заставить пользователя открыть письмо с вирусом, что говорит об эффективности методов социальной инженерии для проведения атак, и о серьезности проблемы [7].

Следует учитывать, что при социоинженерных атаках обнаружение утечки данных для компании происходит после нанесения урона. Поэтому остро стоит проблема анализа поражаемости документов в системе, а также ее пользователей, на предмет социоинженерной атаки, с целью предотвращения риска. [8]

I.3 Постановка цели и задачи

Цель данной ВКР – создание системы, анализирующей защищенность информационных систем от социоинженерных атак посредством учета структуры связи пользователей в системе, степень уязвимости пользователей, расширяя метод Азарова с помощью стохастическо-реляционного подхода.

Для выполнения задачи ставятся следующие задачи:

- Расширить модель предприятия Азарова, основанная на методе Н. В. Хованова определения общей модели измерения ценности, входящие в последовательность «информационная система – персонал – критичные документы»
- Разработать алгоритм имитации социоинженерной атаки злоумышленника на пользователей системы
- Разработать алгоритм вывода меры предосторожностей (защиты): разработать алгоритм поиска самых уязвимых пользователей, хостов и документов

- Реализовать вышеуказанные алгоритмы в программных модулях системы анализа защищенности пользователей от социоинженерных атак

II. II ГЛАВА

II.1 Введение

В данной главе рассматриваются теоретические основания работы, используемые модели, рассмотрены преимущества и недостатки существующих подходов.

II.2 Компоненты комплекса «Информационная система – персонал – критичные документы»

Н.В. Хованов, обобщая и адаптируя в [3] результаты и положения ряда общетеоретических, частных и учебных публикаций, предложил моделировать комплекс «товар – посредник – потребитель» на основе реляционного подхода. Применяя эту модель к предприятию, мы получаем следующие элементы модели [9]:

- Документ – объект, имеющую материальную ценность; конечная цель атакующего.
- Хост – компьютер, с помощью которого пользователь осуществляет взаимодействие с документами.
- Пользователь – человек, работающий на предприятии. Именно с помощью взаимодействия с пользователями атакующий добивается своих целей.
- Атакующий – элемент, лежащий вне предприятия, но необходимый в модели для описания атаки.

II.3 Вероятностно-реляционный подход к задаче

Между элементами модели предприятия существуют связи. В обобщенном, стохастическо-реляционном случае эти связи носят вероятностный характер, имитируя неопределенность и силу этих связей. Следует отметить, что этот

подход включает в себя реляционный детерминированныый подход к задаче: всем связям в таком случае назначается вероятность 1.

Таким образом, мы получили графовую модель предприятия с вероятностными связями.

Для решения задачи имитации атак необходимо определить алгоритм нахождения вероятности доступа атакующего к документу, и поиска наиболее уязвимых документов, пользователей и хостов. Для нахождения вероятности доступа применяется метод, аналогичный используемому в полях Маркова [9] [10]. Так, если атакующий X будет распространять своё взаимодействие через объекты (A_1, A_2, A_3, …, A_N), вплоть до целевого документа D, то вероятность доступа $p_{X,D}$ вычисляется как:

$$P_{X,D} = P_{X,A1}*P_{A1,A2}*P_{A2,A3}*\ldots*P_{AN-1,AN}*P_{AN,D}$$

Где $P_{A,B}$ означает вероятность перехода от узла графа A к узлу графа B.

Однако, следует отметить, что в [9] вероятностный подход рассмотрен не полностью. Не указаны методы поиска наиболее уязвимых элементов, и предполагается, что модель предприятия представляет собой ациклический граф, т.е. Байесову сеть. Также, не указан алгоритм, согласно которому можно осуществить расчет доступа атакующего к документу.

Поиск наиболее уязвимых документов может осуществляться с помощью:

- Оценки вероятности доступа атакующего
- По математическому ожиданию прибыли атакующего от доступа к документу

Поиск наиболее уязвимых пользователей и хостов может осуществляться с помощью оценки вероятности доступа атакующего к пользователю или хосту. Самыми уязвимыми пользователями и хостами являются те, к которым атакующий получит доступ с наибольшей вероятностью.

Таким образом, мы сформировали математическую модель и методы, которыми мы будем оперировать в этой работе.

III. III ГЛАВА

III.1 Введение

Данная глава рассматривает новые элементы модели, введенные автором. Также, приведено полное описание модели предприятия. Кроме этого, данная глава содержит описание и псевдокод используемых алгоритмов.

III.2 Дополнение модели предприятия

Данная работа рассматривает обобщение реляционной модели предприятия на стохастическо-реляционный случай, и без ограничения на отсутствия циклов в графе.

В [9] приводится основы реляционной модели, но описание развито недостаточно (нет подробного описания модели и методов работы с ней), и введено ограничение на наличие циклов в графе. Это ограничение является лишним, если предположить, что атакующий действует оптимально, т.е. достигает документов через путь с наибольшей вероятность, и в пути атакующего не может быть повторяющихся ребер. В таком случае у атакующего будет конечное количество оптимальных путей, так как добавление ещё одного ребра в оптимальный путь уменьшит вероятность доступа, т.е. при изменении ребра путь перестает быть оптимальным.

III.3 Используемая модель предприятия

Опишем получившуюся модель.

Модель предприятия описывает ситуацию, когда атакующий пытается получить доступ к документам, и представляет собой граф, состоит из следующих типов узлов:

- Атакующий
- Пользователь
- Хост (компьютер)

- Документ

Атакующий и пользователь обладают компетенциями [11] – набором определенного числа численных характеристик от 0 до 100, которые задаются экспертно и означают способность воздействовать определенным способом, для атакующего, и слабость к воздействию соответствующим способом, для пользователя.

Документ обладает материальной ценностью, выраженной в денежном эквиваленте. Этот денежный эквивалент – сумма, которую потеряет организация в случае кражи или порчи документа.

Для простоты обращения все объекты проименованы.

Между узлами существуют связи, нагруженные вероятностями. Связи означают, что злоумышленник, получив доступ к одному элементу организации, может получить доступ и к другому. Вероятности означают неопределенность в существовании связи, и что получение доступа может не осуществиться из-за ряда случайных факторов.

Рассмотрим имеющиеся связи предприятия:

- Атакующий-пользователь. Вероятность, присвоенная этой связи, означает то, что атакующему удалось с помощью пользователя воздействовать на предприятие. Эта вероятность зависит от способностей атакующего и пользователя, выраженных в числах – компетенциях. Вероятность воздействия вычисляется как номированный максимум разницы компетенций атакующего к компетенции пользователя. [11] Связи между атакующими не существует, что позволяет одновременно моделировать несколько независимых атак.
- Пользователь-пользователь. Атакующий может распространить своё влияние с одного пользователя на другого; при этом успешность этого

будет зависеть от личных отношений между пользователями. Эта вероятность задается экспертно.

- Пользователь – хост. Для получения доступа к документам атакующий должен попросить пользователя совершить определенные с определенным компьютером. Однако, есть случайные (с точки зрения атакующего) факторы, препятствующие этому: у пользователя может не быть доступа к нему, или компьютер может быть занят. Мы не разделяем эти случаи, а указываем общую вероятность перехода в связи. Вероятность задается экспертно.
- Хост – хост. Так как компьютеры предприятия объединены в локальную сеть, можно с одного компьютера получить доступ к другому. Однако, из-за разграничений прав доступа или из-за особенностей сети этот доступ может быть неосуществим; это выражается в вероятности связи. Вероятность задается экспертно.
- Хост – документ. Получив доступ к нужному хосту, атакующий может с помощью пользователя совершить необходимые действия. Однако, у пользователя может не хватить прав, документ может быть перемещен, и так далее – мы все эти случаи обобщаем под вероятностью перехода. Вероятность задается экспертно.

Также, необходимо разработать алгоритм оценки вероятности доступа к документу. Как уже говорилось, мы предполагаем наихудший случай, что атакующий действует оптимально. Для оценки этой вероятности возможно использовать адаптированный алгоритм поиска кратчайшего пути, который будет учитывать то, что мы оперируем вероятностями, при переходе вероятности должны перемножаться, и задача – найти максимальную вероятность.

Также, необходимо реализовать алгоритм оценки уязвимости остальных узлов. Это также реализуется алгоритмом поиска кратчайшего пути, который кроме итоговой вероятности может указать и вероятности доступа к промежуточным узлам.

Этим требованиям удовлетворяет алгоритм Дейкстры [12]. Поэтому мы будем использовать модификацию этого алгоритма, учитывающую, что мы работаем с вероятностями.

III.4 Псевдокод алгоритма поиска оптимальных путей атак

```
dijkstra(s) =
    for v in V
        d[v] = 0
        used[v] = false
    d[s] = 1
    for i in V
        v = null
        for j in C
            if !used[j] and (v == null or d[j] > d[v])
                v = j
        if d[v] == 0
            break
        used[v] = true
        for e : исходящие из v рёбра
            if d[v] * e.len > d[e.to]
                  d[e.to] = d[v] * e.len
```

IV. IV Глава

IV.1 Введение

Данная глава рассматривает аспекты проектирования архитектуры информационной системы, уточняет интересы сторон и варианты использования.

IV.2 Интересы сторон

Первой стадией разработки системной архитектуры является анализ требований [13]. Рассмотрим интересы сторон, использующих систему, чтобы иметь возможность учесть их интересы при проектировании и разработке.

IV.2.1 Разработчик

Разработчика интересуют параметры системы, связанные с разработкой и поддержкой, такие, как:

1. Сокращение сроков разработки
2. Упрощение поддержки системы

IV.2.2 Аналитик

Аналитик – лицо, непосредственно использующее системы. Его интересуют параметры удобства системы:

1. Скорость запуска системы
2. Субъективное удобство интерфейса
3. Простота ввода исходных данных
4. Простота и наглядность выходных данных

IV.3 Варианты использования системы

Системой непосредственно будет пользоваться аналитик. Необходимо перечислить все варианты использования, чтобы корректно спроектировать

интерфейс, и не упустить вариантов использования системы при разработке и проектировании.

Варианты использования системы аналитиком:

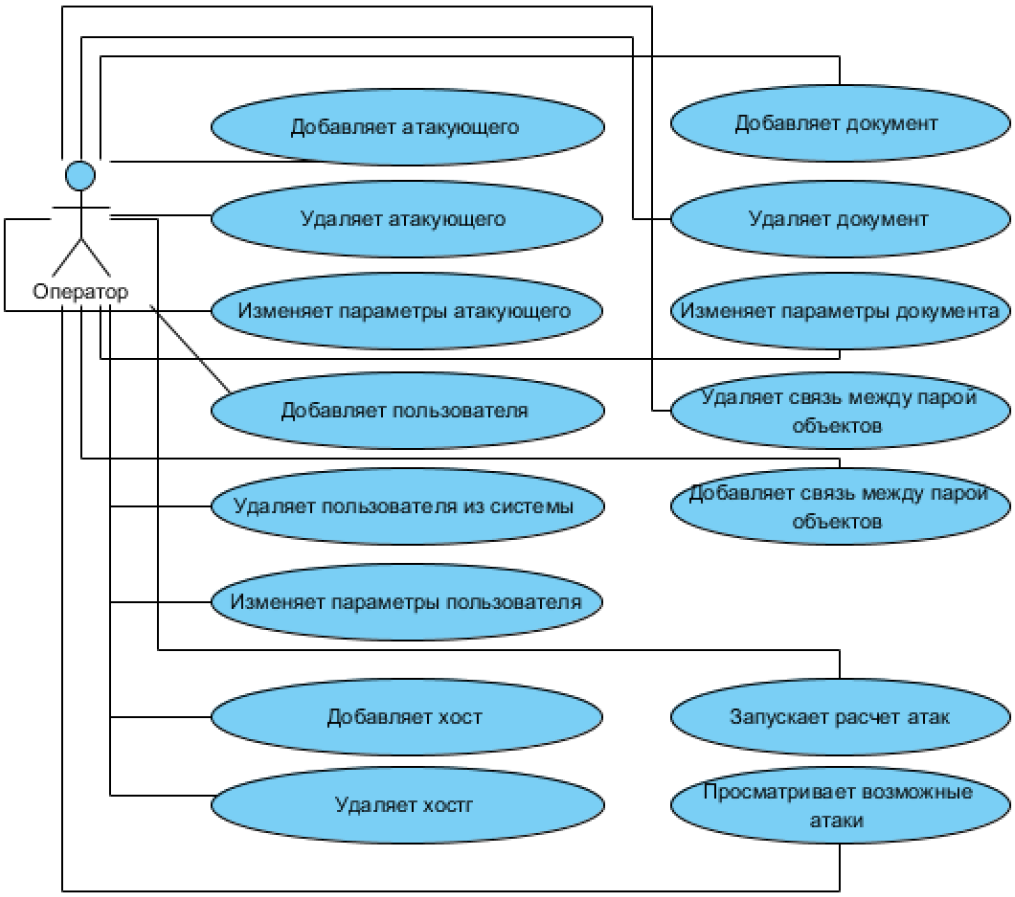


Рисунок 1

IV.4 Выбор технологии

Одним из интересов разработчика является сокращение сроков разработки. Это может достигаться с помощью использования повторных компонент, поэтому по возможности будут использоваться готовые и разработанные решения.

В качестве основных интересов пользователя-аналитика выступает простота интерфейса, так как именно он будет использовать программу по назначению.

В связи с этими требования нам необходимо выбрать язык программирования, который позволит легко создавать пользовательские интерфейсы, содержит большое количество готовых конструкций языка, облегчающих разработку.

Исходя из этого мы выбираем C#, поскольку является объектно-ориентированным языком, удобным для реализации задачи, позволяющим легко создавать пользовательские интерфейсы, и имеющий большой выбор готовых библиотек.

Объектно-ориентированная парадигма позволяет легко описывать модели предприятия. Поскольку у нас уже существует подробное описание модели, выбор объектно-ориентированного подхода в качестве основной парадигмы позволяет сократить трудозатраты на проектирование и реализацию.

IV.5 Проектирование программной архитектуры

Основа программной архитектуры – это классы, представляющие собой элементы математической модели атакуемого предприятия. Для разработки используем метод проектирования «снизу-вверх».

Атакуемое предприятие можно представить в виде направленного графа, где узел – это атакующий, пользователь, хост, или документ. Наличие связи между парой узлов А и Б подразумевает, что действие атакующего может оказать влияние на Б, если будет оказано влияние на А. Связи могут быть нагружены такой дополнительной информацией, как вероятность успешного влияния. Таким образом, создадим класс Edge, который будет хранить информацию о связи между двумя объектами. Чтобы использовать все преимущества языка со строгой типизацией, Edge должен быть шаблонным классом, и содержать три поля: from, to и weight (вес связи).

Следует отметить, что указанные типы узлов формируют иерархию, где связи существуют только в пределах одного уровня, и на уровень ниже (поскольку, например, нет прямых связей между пользователем и документом). Поэтому

класс Node для удобства реализации алгоритма мы тоже можем сделать шаблонным, с двумя параметрами: T и T2. T будет указывать на тип узлов на этом же уровне, T2 – на уровне ниже. Также, добавим методы, которые возвращают списки связей с узлами этого же уровня, и уровня ниже.

Создадим классы Attacker, User, Host и Document, являющиеся расширениями класса Node. Так как для Document не существует класса ниже, шаблонный параметр T2 укажем как Document.

Самый крупный класс – это AttackedOrganization, контейнер, вмещающий в себя все необходимые элементы модели атакуемого предприятия.

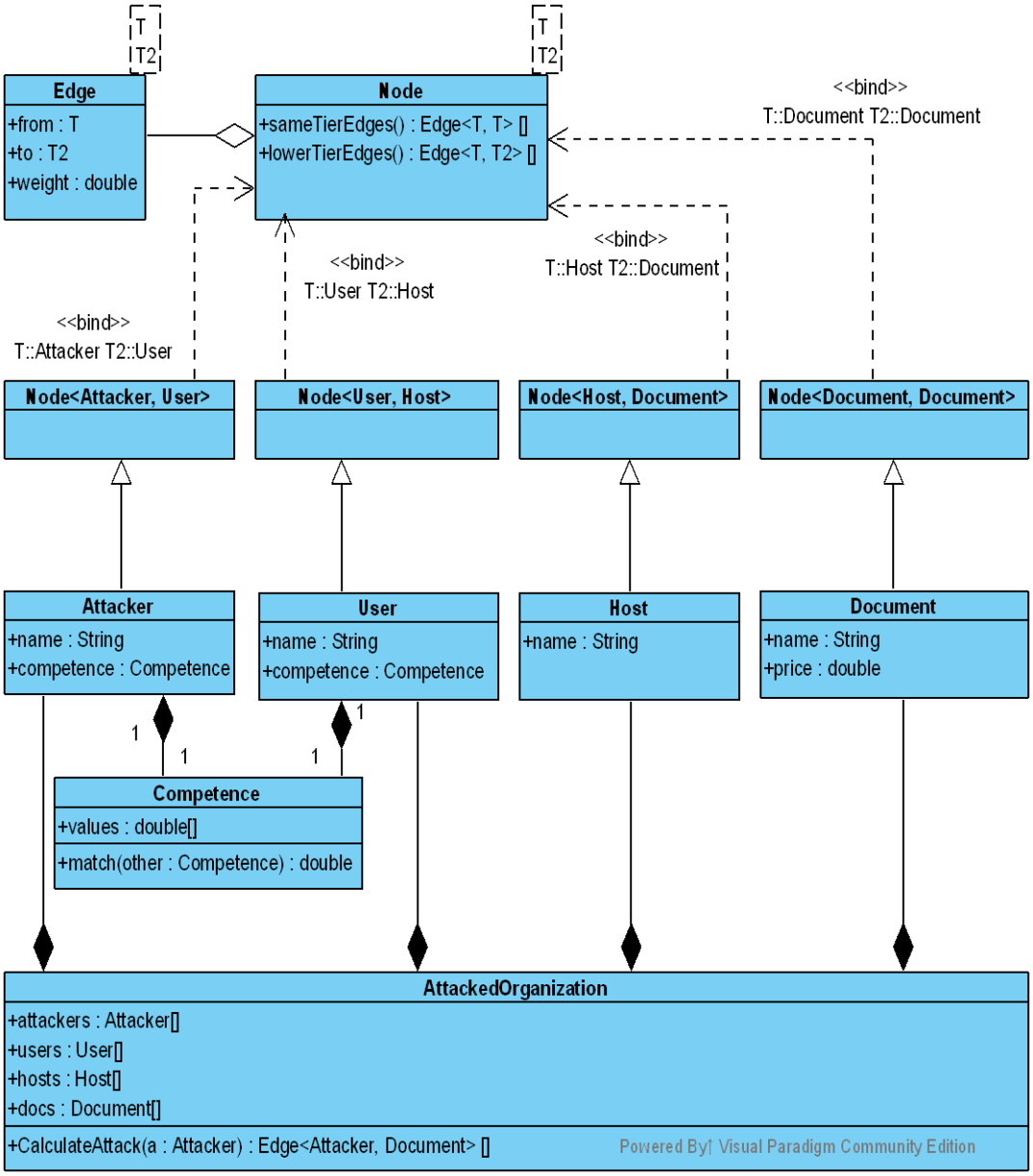


Рисунок 2

IV.6 Выбор инструментария разработки

В качестве среды разработки была выбрана Microsoft Visual Studio 2015, поскольку это последняя на текущий момент версия Visual Studio, интегрированной среды разработки, позволяющей писать на языке C#.

Проектирование, формирование требований и документация архитектуры выполняется с помощью Visual Paradigm, поскольку эта программа является

мощным и бесплатным инструментом, позволяющим выполнить поставленные задачи.

IV.7 Структура хранения данных

Одним из условий удобной работы с программой является сохранение модели атакуемого предприятия между запусками программы.

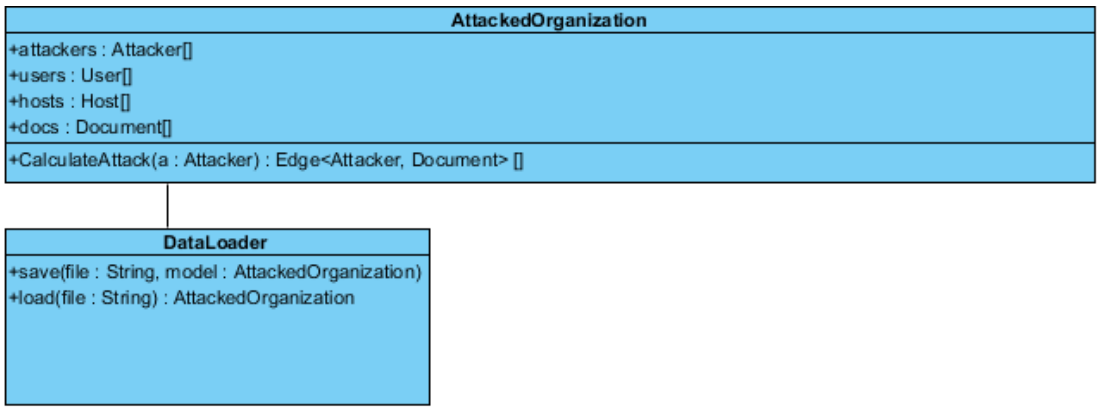
Для реализации такого функционала необходимо разработать структуру хранения данных предприятия, и класс, реализующий сохранение и загрузку модели в спроектированную структуру данных.

Одним из простейших вариантов организации хранения информации является хранение информации в файлах. Этот способ не требует наличия работающей базы данных на локальной машине или выделенном сервере, и достаточно прост.

Для удобства, при запуске программы будет осуществляться чтение файла с моделью из директории с программой. Если файл не был найден, то создается структура предприятия по умолчанию. При закрытии программы модель сохраняется в файл настроек. Файл настроек должен содержать всю необходимую информацию.

Одним из критериев выбора технологии было наличие библиотеки готовых функций. Так, в выбранном нами языке C# существует класс, реализующий необходимый функционал: XmlSerializer [14].

За сохранение будет отвечать класс DataLoader, описание которого представлено на рисунке . Он будет использовать XmlSerializer, чтобы сохранять и загружать модель из файловой системы.



IV.8 Проектирования пользовательского интерфейса

Для отображения модели в удобном виде, а также простого доступа пользователю к функциям, необходимо разработать пользовательский интерфейс. Так как аналитик будет взаимодействовать с программой через интерфейс, его основным требованием является простота и интуитивность графического интерфейса.

Одним из способов проектирования пользовательских интерфейсов является создание зарисовок [15]. Поэтому для доступа к необходимому функционалу с помощью был создан набор зарисовок пользовательского интерфейса. Рисунок 3 содержит зарисовку интерфейса основного окна, а рисунок 4 содержит зарисовку интерфейса диалога редактирования связей, который открывается при нажатии на кнопку редактирования связей на основном окне.

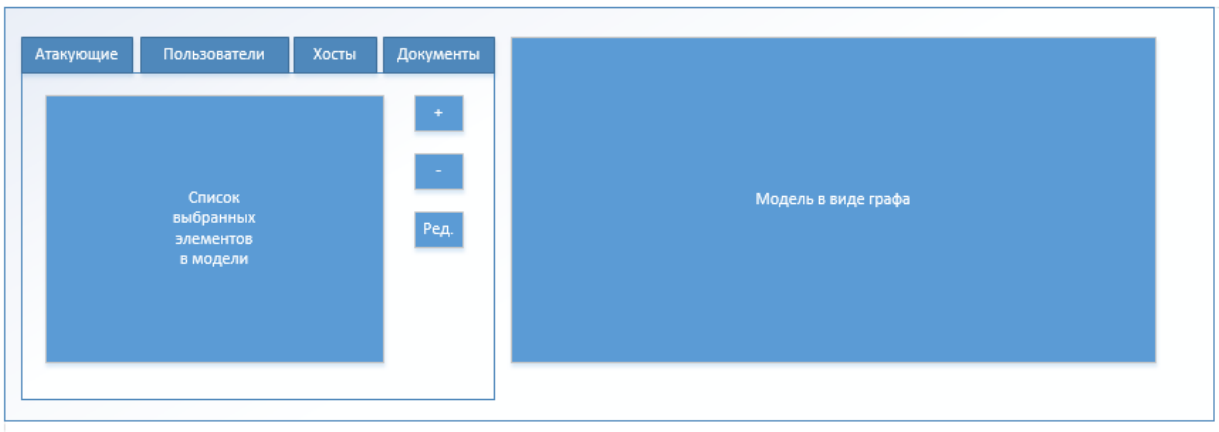


Рисунок 3. Интерфейс основного окна

Рисунок 4. Интерфейс окна редактирования связей

V. V ГЛАВА

V.1 Введение

Данная глава рассматривает аспекты и особенности реализации спроектированной программной архитектуры модели предприятия, вспомогательных классов и пользовательского интерфейса.

V.2 Реализация программной архитектуры модели

V.2.1 AttackerOrganization

Этот класс содержит функцию CalculateAttack, принимающий объект-атакующего, рассчитывающий возможные атаки, и возвращающий список ребер, содержащих возможные атакуемые документы, и вероятность их успешной атаки. Также, этот класс предоставляет открытый доступ к спискам атакующих, пользователей, нападающих, хостов и документов, позволяя удалять и добавлять элементы модели предприятия. Кроме того, именно этот класс содержит необходимые вспомогательные функции для моделирования атаки.

V.2.2 Attacker

Класс атакующего предоставляет открытый доступ к своему имени. Также, методы AddLink и UpdateLinks позволяют добавлять одну связь и изменять все связи разом.

V.2.3 User

Класс пользователя предоставляет открытый доступ к своему имени. Также, методы AddLink и UpdateLinks позволяют добавлять одну связь и изменять все связи разом. Методы перегружены, позволяя либо добавлять связь с другим пользователем, либо с хостом. В случае добавления связи с пользователем эта связь автоматически добавляется и в список связей другого пользователя. При

вызове UpdateLinks все связи других пользователей с текущим удаляются, и заменяются на новые.

V.2.4 Host

Класс хоста предоставляет открытый доступ к своему имени. Также, методы AddLink и UpdateLinks позволяют добавлять одну связь и изменять все связи разом. Методы перегружены, позволяя либо добавлять связь с другим хостом, либо с документом. В случае добавления связи с хостом эта связь автоматически добавляется и в список связей другого хоста. При вызове UpdateLinks все связи других хостов с текущим удаляются, и заменяются на новые.

V.2.5 Document

Класс предоставляет открытый доступ к полям, обозначающим его имя и стоимость. Документы не имеют связей ни между собой, ни с низлежащим уровнем иерархии, поэтому вызов методов getSameTierEdges и getLowerTierEdges вызывают исключение.

V.3 Реализация пользовательского интерфейса

Как отмечается в [15], проектирование – итеративный процесс, в частности, при реализации спроектированного пользовательского интерфейса возможно внесение дополнительных уточнений по запросу пользователя или из-за специфики/объема отображаемых данных.

Одной из особенностей является отображение модели предприятия. Необходимо нарисовать модель предприятия в виде графа таким образом, чтобы получилась наглядная и понятная картинка.

При отрисовке графа будем отображать узлы в виде кружков с названием объекта, а связи – прямыми линиями. Цвет линий будет соответствовать силе связи:

- 0: зеленый цвет
- 0,5: черный цвет
- 1: красный цвет

Переходные значения будут линейно интерполироваться.

Нетривиальной задачей является размещение узлов графа на двумерной плоскости (т.е. планаризация) таким образом, чтобы результат был максимально читаем [16]. Существует множество разных алгоритмов, использующих различные эвристики [16]. Одним из простейших вариантов планаризации является расположение узлов по кругу. Мы на основе этого алгоритма можем создать модификацию, которая будет учитывать, что узлы графа образуют иерархию, и располагать узлы каждого уровня по отдельному кругу. Мы предлагаем подбирать центры кругов и их радиусы таким образом, чтобы радиус был пропорционален количеству элементов на этом уровне.

Псевдокод планаризации графа (Va – вершины атакующих, Vu – вершины пользователей, Vh – вершины хоста, Vd – вершины документов, V – все вершины, x[] – ассоциативный массив, сопоставляющий вершине её координату X на поле для отображения, y[] – ассоциативный массив, сопоставляющий вершине её координату Y на поле для отображения, W,H – ширина и высота поля для отображения соответственно):

```
planarize(Va, Vu, Vh, Vd, W, H) =
      count = |Va| + |Vu| + |Vh| + |Vd|
      radiusa = W * |Va| / count
      radiusu = W * |Vu| / count
      radiush = W * |Vh| / count
      radiusd = W * |Vd| / count
      i = 0
      for a in Va
            fi = 2 * Pi * i / |Va|
            x[a] = cos(fi) * radiusa
            y[a] = sin(fi) * H / 2
            i = i + 1
      i = 0
```

```
for u in Vu
      fi = 2 * Pi * i / |Vu|
      x[a] = cos(fi) * radiusu + radiusa * 2
      y[a] = sin(fi) * H / 2
      i = i + 1
i = 0
for h in Vh
      fi = 2 * Pi * i / |Vh|
      x[a] = cos(fi) * radiush + radius * 2 + radiusa * 2
      y[a] = sin(fi) * H / 2
      i = i + 1
i = 0
for d in Vd
      fi = 2 * Pi * i / |Vd|
      x[a] = cos(fi) * radiusd + radiush*2+ radius*2 + radiusa * 2
      y[a] = sin(fi) * H / 2
      i = i + 1
return {x, y}
```

VI. VI ГЛАВА

VI.1 Введение

Данная глава содержит описание пользовательского интерфейса, и описание действий, которые необходимо предпринять пользователю, чтобы осуществить необходимые действия по вводу данных или расчету.

VI.2 Описание интерфейса

Основное окно

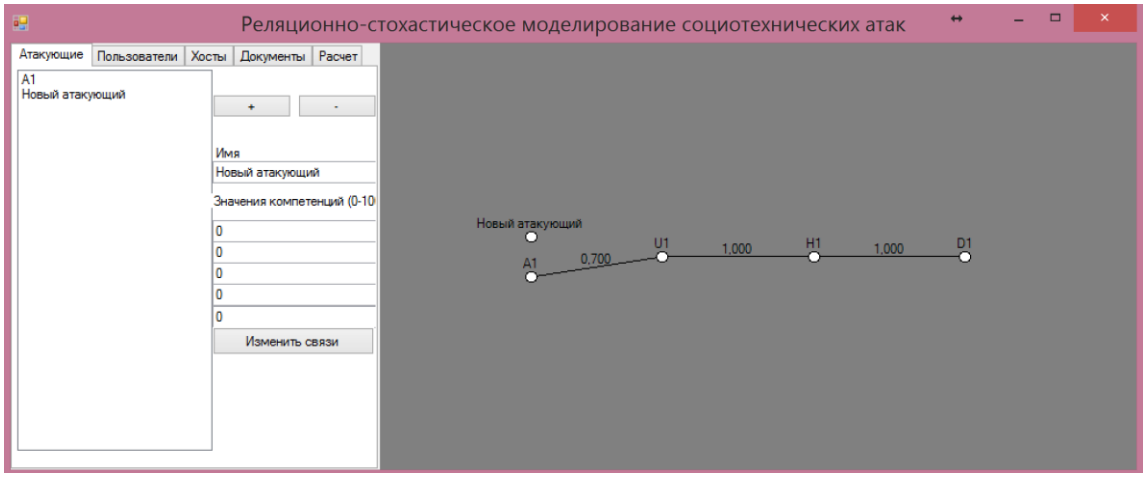


Рисунок 5

В правой части основного окна представлена визуализация графа предприятия. Слева направо, она содержит: узлы атакующих, узлы пользователей, узлы хостов, узлы документов. Узлы подписаны именами для удобства. Также, визуализация содержит связи с подписью значения веса.

Окно редактирования связей

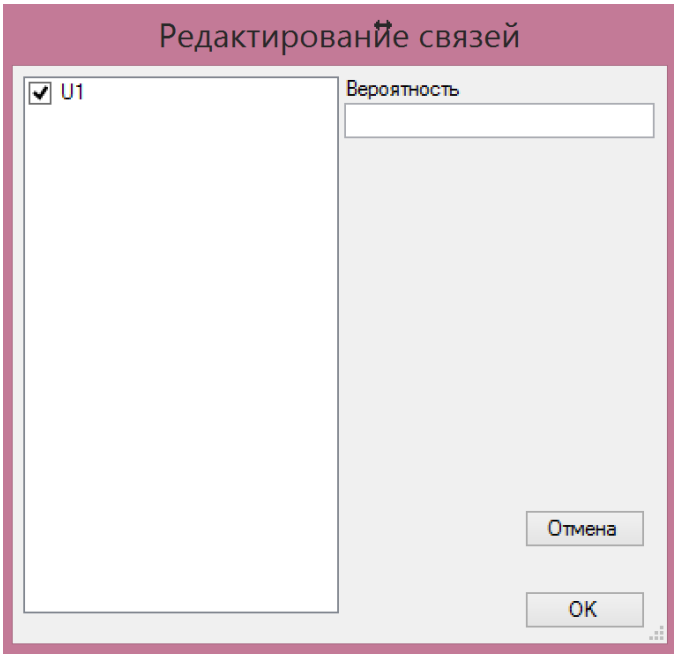


Рисунок 6

Данное окно позволяет добавлять и удалять связи между компонентами. В левой части содержится список доступных компонент для связи. В правой части содержатся

VI.3 Описание типовых действий

VI.3.1 Добавление элемента модели

Для добавления необходимого элемента модели необходимо выбрать соответствующую вкладку в левой части основного окна, и нажать на кнопку «+».

VI.3.2 Удаление элемента модели

Для удаления необходимого элемента модели необходимо выбрать соответствующую вкладку в левой части основного окна, выбрать удаляемый элемент в списке, и нажать на кнопку «-».

VI.3.3 Редактирование свойств элемента модели

Для редактирования необходимого элемента модели необходимо выбрать соответствующую вкладку в левой части основного окна, и выбрать редактируемый элемент в списке. При этом его редактируемые поля появятся правее списка. Далее, необходимо внести изменения в текстовые поля справа от списка элементов.

VI.3.4 Добавление связи между элементами модели

Для добавления связи между элементами модели необходимо выбрать соответствующую вкладку в левой части основного окна, и выбрать первый элемент в списке. После этого необходимо нажать кнопку редактирования связей, тип которой зависит от типа второго элемента. Откроется окно редактирования связей. Необходимо найти второй элемент в списке, и поставить галочку слева от него, и занести экспертную вероятность связи в текстовое поле в правой части окна. После этого необходимо нажать кнопку «ОК».

VI.3.5 Удаление связи между элементами модели

Для удаления связи между элементами модели необходимо выбрать соответствующую вкладку в левой части основного окна, и выбрать первый элемент в списке. После этого необходимо нажать кнопку редактирования связей, тип которой зависит от типа второго элемента. Откроется окно редактирования связей. Необходимо найти второй элемент в списке, и снять галочку слева от него. После этого необходимо нажать кнопку «ОК».

VI.3.6 Запуск моделирования атаки и просмотр результатов

Для запуска моделирования необходимо выбрать вкладку «Расчет», и нажать на кнопку «Рассчитать». При этом в списке в левой части окна появятся вероятности доступа к документам каждого атакующего, и математическое

ожидание экономических потерь. Второй список содержит список наиболее уязвимых узлов, упорядоченных по вероятности поражения.

VII. Список литературы

[1] А. Батогов, «В 2014 ГОДУ ЗНАЧИТЕЛЬНО УВЕЛИЧИЛОСЬ КОЛИЧЕСТВО DDOS-АТАК,» [В Интернете]. Available: http://hi-news.ru/internet/v-2014-godu-znachitelno-uvelichilos-kolichestvo-ddos-atak.html. [Дата обращения: 15 март 2016].

[2] C. W. Flink, «Weakest Link in Information System Security,» *WAEPSSD*, 2002.

[3] Н. В. Хованов, «Общая модель измерения ценности экономических благ,» *Применение математики в экономике. Вып. 18,* pp. 108-134, 2009.

[4] С. В. Вихорев, «Угрозы информационной безопасности,» *Сетевые атаки и угрозы информационной безопасности,* 2001.

[5] «ВОЗМОЖНЫЕ УГРОЗЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ И ИХ СПЕЦИФИКА,» [В Интернете]. Available: http://www.e-nigma.ru/stat/dip_2/. [Дата обращения: 2016 03 15].

[6] «Как защитить внутреннюю сеть и сотрудников компании от атак, основанных на использовании социотехники,» [В Интернете]. Available: https://technet.microsoft.com/ru-ru/library/cc875841.aspx. [Дата обращения: 15 3 2016].

[7] «Социальная инженерия,» [В Интернете]. Available: http://www.kaspersky.ru/internet-security-center/threats/malware-social-engineering. [Дата обращения: 22 03 2016].

[8] «Социальная инженерия, или Как «взломать» человека,» [В Интернете]. Available: https://blog.kaspersky.ru/socialnaya-inzheneriya-ili-kak-vzlomat-cheloveka/2559/. [Дата обращения: 22 03 2016].

[9] А. А. Азаров, Т. В. Тулупьева и А. Л. Тулупьев , «Агентоориентированный подход к моделированию комплекса «Информационная Система –Персонал – Злоумышленник»в задачах оценки защищенности от социоинженерных атак.,» *Список-2012: Материалы всероссийской научной конференции по проблемам информатики,* p. 374–377, 2012.

[10] В. Ф. Мусина, «Байесовские сети доверия как вероятностная графическая модель для оценки экономических рисков,» *Труды СПИ-ИРАН,* № 25, 2013.

[11] А. А. Азаров, Т. В. Тулупьева, А. А. Фильченков, А. В. Суворова, В. Ф. Мусина и А. Л. Тулупьев, «Защищенность пользователей информационных систем от социоинженерных атак: психологические аспекты,» p. 207–218, Материалы Второй Международной научно-практической конференции "Социальный компьютинг, технологии развития, социально-гуманитарные эффекты".

[12] А. В. Левитин, «Алгоритмы. Введение в разработку и анализ,» 2006.

[13] ГОСТ 34.601-90 АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ. СТАДИИ СОЗДАНИЯ., 1992.

[14] «Примеры XML-сериализации,» [В Интернете]. Available: https://msdn.microsoft.com/ru-ru/library/58a18dwa(v=vs.100).aspx. [Дата обращения: 26 03 2016].

[15] «User Interface (UI) Prototypes: An Agile Introduction,» [В Интернете]. Available: http://agilemodeling.com/artifacts/uiPrototype.htm. [Дата обращения: 20 03 2016].

[16] R. Tamassia, «Crossings and planarization,» в *Crossings and Planarization*, 2013, pp. 43-85.

[17] R. Jayakumar и K. Thulasiraman, «O(n^2) Algorithms for Graph Planarization,» *IEEE Transactions on computer-aided design,* т. 8, № 3, pp. 257-269, 1989.